Adaptation of Population Sizes by Competing Subpopulations

Dirk Schlierkamp-Voosen
GMD - German National Research Center
for Information Technology
D-53754 Sankt Augustin, Germany
schlierkamp-voosen@gmd.de

Heinz Mühlenbein
GMD - German National Research Center
for Information Technology
D-53754 Sankt Augustin, Germany
muehlenbein@gmd.de

***Abstract*— We present a competiton scheme which dynamically allocates the number of trials given to different search strategies. The competition scheme changes the sizes of the subgroups, but also the size of the whole population. The competition scheme is able to combine the strengths of individual search strategies in a synergetic way. This claim is demonstrated by numerical experiments with two difficult functions to be optimized.**

***KeyWords*— breeder genetic algorithm, population dynamics, competition, consumption factor, variable population size**

I. Introduction

Most evolutionary algorithms depend on a set of control parameters. Often the optimal setting of the parameters depends on the particular application. Moreover the optimal control parameters may vary for different stages of the search.

There are two basic ways of adapting the parameters. The first is to use some externally specified schedule [7] [5]. The second applies the mechanisms of evolution itself. This approach is successfully used in evolution strategies [2].

In this paper we present an adaptation method based on competing subpopulations. It is a major extension of the scheme presented in [11]. The new competition allows the adaptation of one of the most important control parameters of evolutionary algorithms: the *size of the population.*

The outline of this paper is as follows. In section 2 ecological models relevant to our competition scheme are presented. In section 3 the competition scheme for evolutionary algorithms is explained. The most important operators and control parameters of the breeder genetic algorithms (BGA) are summarized in section 4. The performance of the competition is shown in section 5 for multimodal functions. A conclusion is given in section 6.

II. The ecology of populations

Nature has managed to arrive at very complex systems using genetic evolution by natural selection. This mechanism has been applied to artificial systems by standard evolutionary algorithms. But nature does not stop with just one species. It has created many species which interact in ecologies. By the interaction of different species their population size is continuously adapted. The success of these mechanisms suggests that they might also be relevant for extending evolutionary algorithms. Therefore a short overview of these mechanisms seems appropriate.

Ecology deals with the relationships between organisms and their environment, including in particular the interactions between different species. Unfortunately even the analysis of the interaction between two species can be quite complicated, involving the effects of exterior and interior parameters. As a first approximation one may distinguish four basic situations — *competition, predator-prey, symbiosis* and *host-parasite.*

The most popular equations for analyzing the interaction of species are generalizations of the famous Lotka-Volterra equations. The Lotka-Volterra competition equation describes the development of competing species as follows [3]

$$\frac{dN_i}{dt} = r_{i0} \cdot N_i \left(1 - \frac{N_i + \alpha_{ij} N_j}{K_i}\right) \qquad (1)$$

Here N_i denotes the *population size* of species i, r_{i0} the *intrinsic rate of increase*, K_i the *carrying capacity* and α_{ij} the *interaction coefficient* to species j. The carrying capacity is the maximal number of individuals which can be supported by the limiting resource.

This equation has been studied extensively [3]. It is very useful for understanding the complex patterns which may arise by interacting species. For a competition scheme to be implemented these equations cannot be used because the interaction coefficients cannot be specified in advance. Therefore our model omits the interaction coefficients and uses a quality criterion to adapt the population sizes of the competing subpopulations.

III. Competition between subpopulations

A. The basic competition model

Our competition scheme requires a *quality criterion* to rate a group, a *gain criterion* to reward or punish the groups, an *evaluation interval*, and a *migration interval.* The evaluation interval gives each strategy the chance to demonstrate its performance in a certain time window. By occasional migration of the best individuals groups which performed badly are given a better chance for the next competition. The sizes of the groups have a lower limit. Therefore no strategy is lost. The number of competing subpopulations (S) depends on the set of strategies used in the competition. Normally the number of groups is between 2 and 8.

Our **quality criterion** (Q) is based on the fitness of the best individual of the group. To avoid an inefficient oscillation of group sizes we use information about the last ω generations for the evaluation. The vector $\vec{b} = (b_1, \ldots, b_\omega)^T$ provides the indices of the groups which contained the global best individual in the last ω generations. $b_k \in \{1,\ldots,S\}$ with $k=1,\ldots,\omega$ denotes the index of the group which contained the best individual $k-1$ generations before. So $k=1$ denotes the current generation, $k=2$ the previous one, etc.. In our experiments the time window ω is set to 10.

The following formula describes the quality of group i.

$$Q_i(\vec{b}) = \sum_{k=1}^{\omega} \begin{cases} \frac{\omega-k+1}{\omega} & : \quad i = b_k \\ 0 & : \quad i \neq b_k \end{cases} \tag{2}$$

In the following w^* denotes the index of the group with highest quality. If several groups have the best quality, one of them is randomly chosen.

The **gain criterion** (G) defines how the population size of each group is modified according to its quality. Normally, the size of the group with the best quality increases, the sizes of all other groups decrease. The following scheme increases the size of the best group by the accumulated loss of the others. The loss of a group is proportional to its population size. The loss factor $\kappa \in [0,1]$ defines the rate of loss. The extinction of a group is avoided by a spezified minimal popsize $N_i^{\min}$.

The decrease of populations size L_i^t is determined by the following equation:

$$L_i^t = \begin{cases} 0 & : i = w^* \\ N_i^t \cdot \kappa & : i \neq w^* \;\wedge\; N_i^t(1-\kappa) \geq N_i^{\min} \\ N_i^t - N_i^{\min} & : i \neq w^* \;\wedge\; N_i^t(1-\kappa) < N_i^{\min} \end{cases} \tag{3}$$

for $i = 1,\ldots,S$. Here N_i^t denotes the size of group i in generation t. Typically we use a loss factor $\kappa = 0.125$.

The difference of population size ΔN_i^t is given by the following equation. The winner species gets the accumulated loss of all other species.

$$\Delta N_i^t = \begin{cases} \sum_{j=1}^{S} L_j^t & : \quad i = w^* \\ -L_i^t & : \quad i \neq w^* \end{cases} \tag{4}$$

The population size of each group of the next generation is given by:

$$N_i^{t+1} = N_i^t + \Delta N_i^t \tag{5}$$

This gain criterion leads to a fast adaptation of the group sizes. Each group, exept the best, loses the same percentage of individuals.

The **evaluation interval** η and the **migration interval** θ are rather robust external parameters. The evalutation interval specifies the number of generations between successive competitions. The migration interval defines the number of generations between a migration of the global best individual to the other groups. After a migration all subpopulations contain the global best individual. Typically we work with $\eta = 4$ and $\theta = 16$.

If one compares equations (3) and (4) with the generalized Lotka-Volterra equation (1) the following major difference can be observed. Our equations are linear whereas the Lotka-Volterra equation contains the nonlinear term $N_i \cdot N_j$. The reason for this difference is that the Lotka-Volterra equation models competition by individual encounters of two of each species. Therefore the reduction of a species depends on the population sizes of the competing species. In contrast, our competition scheme evaluates whole groups by taking the best individual as evaluation criterion.

The current competition scheme seems appropriate for cases in which the strategies used by the different groups differ substantially. If the different strategies are more similar to each other, then a competitor scheme as modeled by the Lotka-Volterra equation might be better. A step in this direction is the extended competition model where the size of the total population may vary.

B. The extended competition model

If search strategies differ very much they may also require a different population size to be efficient. It has been shown in [9] that mutation is most efficient in small populations whereas recombination needs a larger population size. This can be modeled by introducing growth rates which depend on the group.

In our implementation we introduced a *consumption factor* γ for each subpopulation. Biologically a consumption factor specifies the consumption of the limited resource by one individual of a species – the higher the consumption factors the lower the number of individuals which can be supported by that resource.

We implemented this extension by introducing a *normalized population size* $\tilde{N}$.

$$\tilde{N}_i = \gamma_i \cdot N_i \tag{6}$$

The gain criterion of equation (4) is now applied to the normalized population sizes. The sum of the normalized population sizes remains constant because it is limited by the limited resource K. For $\gamma_i = 1.0$ $(\forall i = 1,\ldots,S)$ the extended model is the same as the basic model.

$$\sum_{i=1}^{S} \tilde{N}_i = K \tag{7}$$

In contrast to the basic competition model the sum of the real population sizes varies during a simulation. This extended competition scheme is very useful for multimodal problems where it is useful to locate the region of attraction of the global (or a good local) optimum by a breadth search and to do the fine adaptation by an exploring strategy afterwards. In this case the strategy performing breadth search gets a lower γ than the other strategy. So the total population size is high at the beginning when the breadth search works and low at the end when the fine adaption is done. So the whole population size is adapted during the run by the competition model.

IV. The BGA for continuous parameter optimization

Let an unconstrained optimization problem be given on a domain $D \subset \mathbb{R}^n$

$$\min(F(\vec{x})) \qquad a_i \leq x_i \leq b_i \qquad i = 1, \ldots, n \ . \tag{8}$$

The breeder genetic algorithm **BGA** was designed to solve the above problem [8]. Like evolution strategies it uses a real-coded representation. The selection is done by *truncation selection*, also called mass selection by breeders. In this selection, which is identical to the (μ, λ)-selection in evolution strategies, the $T\%$ best of the individuals are selected as parents and then mated randomly. Normally we use the elitist version of truncation selection where the best individual always survives. The BGA uses a set of search strategies. Here a search strategy of an EA is defined as an arrangement of operators and their external parameters. An overview of search strategies based on recombination can be found in [13] and [12]. In [13] it was shown that a new recombination scheme called **fuzzy recombination** (FR) worked best as a breadth search. In this paper we only describe the BGA mutation scheme now in use and the BGA line recombination which also uses this scheme.

BGA mutation $\mathsf{BM}(\rho, k, \nu)$

The current BGA mutation scheme depends on the *mutation range* ρ, the *precision factor* k and a new parameter ν which is used to modify more than one variable. ν will be described later. The standard BGA mutation randomly selects just one of the variables $x_1, \ldots, x_n$.

Given x_i a new value z_i is computed according to

$$z_i = x_i + \rho_i \cdot \delta_i(k) \tag{9}$$

ρ_i is normally set to the domain of definition of variable x_i ($\rho_i = b_i - a_i$). $\delta(k)$ is a random variable which is computed as follows:

$$\delta(k) = \mathrm{sign}(u) \cdot 2^{-k \cdot |u|} \qquad (k \geq 0) \tag{10}$$

with $u \sim \mathbf{U}([-1, 1])$. $\mathbf{U}([v, w])$ denotes the uniform probability distribution with support $(v, w) \subset \mathbb{R}$.

k is called the precision constant. The smallest absolute value of $\delta(k)$ is 2^{-k} the highest value is 1.0. Therefore the step sizes of the BGA-mutation are contained in the interval $[\rho_i \cdot 2^{-k}, \rho_i]$.

The pdf of the random variable $\delta(k)$ is:

$$f_k(x) = \begin{cases} -\frac{1}{2 \cdot k \cdot \ln 2} \cdot \frac{1}{x} & : \quad -1 \leq x \leq 2^{-k} \\ \frac{1}{2 \cdot k \cdot \ln 2} \cdot \frac{1}{x} & : \quad 2^{-k} \leq x \leq 1 \\ 0 & : \quad else \end{cases} \tag{11}$$

Figure 1 shows the pdf of $\delta(k)$ for various k.

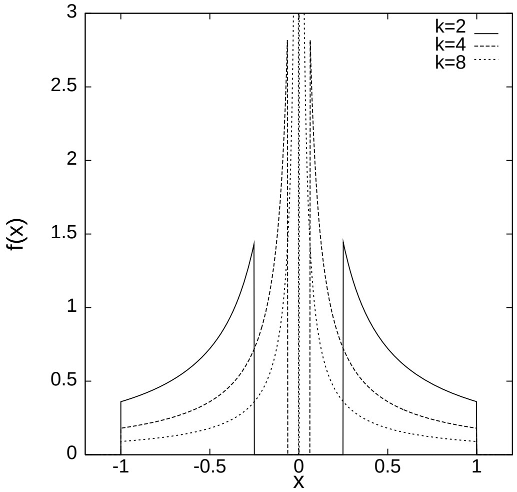

Fig. 1. Density function of the random variable $\delta(k)$ for k=2,4 and 8.

This mutation scheme is based on the discrete BGA mutation scheme introduced in [8]. The rationale of the new continuous mutation scheme is similar to that of the discrete one which has been explained in [8]. Because of generating much more small values this mutation scheme is very robust according to the step size factor ρ. Whereas the standard BGA uses one fixed ρ which is set to domain of definition of the variables, a competition between subpopulations equipped with different ρs is used for problems whose variables are not constrained to finite intervals.

An extension of the BGA-mutation is specified by a third parameter ν. Whereas the former BGA-mutation applied the mutation to each object variable with a probability of $p_m = 1/n$, the new BGA-mutation modifies all object parameters in the following way: The object parameters are modified in a random order denoted by $\vec{\iota} = (\iota_1, \iota_2, \ldots, \iota_n)$. The components of $\vec{x}$ are modified by:

$$z_{\iota_j} = x_{\iota_j} + \rho_{\iota_j} \cdot \delta_{\iota_j}(k) \cdot \nu^{j-1} \qquad j = 1, \ldots, n \tag{12}$$

The higher the rank of a variable the smaller is the change (for the same ρ). For the standard BGA-mutation, where exact one variable is modified, we set $\nu = 0$ (note: $0^0 = 1$). The mutation stepsize decreases exponentially starting from variable x_i. This is in accordance with the design rationale of the BGA mutation.

A variant of this mutation scheme, denoted by $\mathsf{BM}^{+}(\rho, k, \nu)$, does not work with a random ordered $\vec{\iota}$, but sorts the entries of $\vec{\iota}$ relative to a random chosen element of $\vec{x}$. Let i be the index of this random chosen element. Then the $\vec{\iota}$ is set as: $\vec{\iota} = (i, i-1, i+1, i+2, i-2, \ldots)$. In contrast to the standard BGA-mutation the random variable $\delta(k)$ is sampled only once per creation of an offspring.

$$z_{\iota_j} = x_{\iota_j} + \rho_{\iota_j} \cdot \delta(k) \cdot \nu^{j-1} \qquad j = 1, \ldots, n \tag{13}$$

The higher the distance to i the smaller is the change. This variant is suited for fitness functions where neighbored variables are strongly correlated. It was successfully applied to problems of variational calculus.

BGA line recombination $\mathsf{BLR}(\rho, k)$

The BGA line recombination uses components from both, mutation and recombination. It creates new points in a direction given by the two parent points. The placement of the point is done by the BGA mutation scheme. It works as follows: Let $\vec{x} = (x_1, \ldots, x_n)$ and $\vec{y} = (y_1, \ldots, y_n)$ be the parent strings with $\vec{x}$ being the one with better fitness. Then the offspring $\vec{z} = (z_1, \ldots, z_n)$ is computed by

$$z_i = x_i + \rho_i \cdot \delta(k) \cdot \frac{y_i - x_i}{||\vec{x} - \vec{y}||} \qquad i = 1, \ldots, n \tag{14}$$

where $\delta(k)$ is realized as defined in equation (10).

In contrast to standard recombination operators the step size is independent of the distance of the parents. So the BGA line recombination may generate points which are far from the given points $\vec{x}$ and $\vec{y}$.

V. Numerical results

In [11] the efficiency of the competition concept was demonstrated by adapting the step size of the BGA-Mutation to the current situation. In this paper the robustness of the competition concept is demonstrated with two functions, both of which are very difficult to optimize: Rosenbrock's function [10] and Fletcher and Powell's function [4].

Whereas Rosenbrock's function can be characterized as a steep curved valley, Fletcher and Powell's function is based on trigonometric functions and is multimodal with the minima randomly distributed. The locations of the minima are defined by randomly generated matrices. In our experiments we used the matrices generated by Bäck [1].

F_{ROS} is a generalization of Rosenbrock's function to n variables.

$$F_{\text{ROS}}(\vec{x}) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right) \tag{15}$$

The definition range is $-5.12 \leq x_i \leq 5.12$. The optimum at $\vec{x}^* = (1, \ldots, 1)$ has the function value $F^*_{\text{ROS}} = 0.0$.

The definition of Fletcher and Powell's function is given by:

$$\begin{aligned} F_{\text{FP}}(\vec{x}) &= \sum_{i=1}^{n} (A_i - B_i)^2 \qquad (16) \\ A_i &= \sum_{j=1}^{n} (a_{ij} \sin \alpha_j + b_{ij} \cos \alpha_j) \\ B_i &= \sum_{j=1}^{n} (a_{ij} \sin x_j + b_{ij} \cos x_j) \end{aligned}$$

with $a_{ij}, b_{ij} \in [-100, 100]$ and $\alpha_i \in [-\pi, \pi]$.

The definition range is $-\pi \leq x_i \leq \pi$. The optimum at $\vec{x}^* = \vec{\alpha}$ has the function value $F^*_{\text{FP}} = 0.0$.

A. Experiments

The experiments specified in table 1 and table 3 demonstrate both the efficiency and the robustness of the competition concept. The first four experiments are typical BGA runs without competition. Experiment 1 performs a breadth search by using the fuzzy recombination (FR) with a high population size ($N = 512$) and a low selection intensity ($T = 0.5$). Experiment 2 and 3 use the BGA-mutation (BM$^+$) with different population sizes. Experiment 4 combines both operators in one strategy.

The last three experiments use the competition model. In experiments 5 and 6 the basic competition model is used with four subpopulations with different mutation ranges ρ and a reduced precision factor k. Experiment 5 has the same operators and total population size as experiment 2. The same is true for experiments 6 and 3.

Experiment 7 uses the extended competition model introduced in this paper. Here a competition between the breadth search strategy of experiment 1 and the exploring search of experiment 2 and 3 is combined.

E	S	i	N_i^0	γ_i	OP		T
					FR	BM$^+$	
1	1	1	512	1.0	FR	–	0.5
2	1	1	16	1.0	–	BM$^+(\rho, 32, \nu)$	0.25
3	1	1	64	1.0	–	BM$^+(\rho, 32, \nu)$	0.25
4	1	1	512	1.0	FR	BM$^+(\rho, 32, \nu)$	0.5
5	4	1	10	1.0	–	BM$^+(\rho, 5, \nu)$	0.25
		2	2	1.0	–	BM$^+(\rho \cdot 2^{-1\cdot 5}, 5, \nu)$	0.25
		3	2	1.0	–	BM$^+(\rho \cdot 2^{-2\cdot 5}, 5, \nu)$	0.25
		4	2	1.0	–	BM$^+(\rho \cdot 2^{-3\cdot 5}, 5, \nu)$	0.25
6	4	1	52	1.0	–	BM$^+(\rho, 5, \nu)$	0.25
		2	4	1.0	–	BM$^+(\rho \cdot 2^{-1\cdot 5}, 5, \nu)$	0.25
		3	4	1.0	–	BM$^+(\rho \cdot 2^{-2\cdot 5}, 5, \nu)$	0.25
		4	4	1.0	–	BM$^+(\rho \cdot 2^{-3\cdot 5}, 5, \nu)$	0.25
7	2	1	512	0.0625	FR	–	0.5
		2	4	1.0	–	BM$^+(\rho, 32, \nu)$	0.25

Table 1. Experiments for Rosenbrock's function. E denotes the index of the experiment, S the number of subpopulations, i the index of a subpopulation and γ_i the corresponding consumption factor. N_i^0 is the initial size of a subpopulation. BM$^+(\rho, k, \nu)$ denotes the BGA-mutation and FR the fuzzy recombination (default: $\rho = 10.24$). T is the truncation threshold. ν is set to 0.5.

B. Development of population sizes

The development of populations sizes during a competition run is shown in Figures 2 and 3.

Figure 2 shows a basic competition between four subpopulations with different step sizes analogue to experiment 6. At the beginning the subpopulation equipped with the highest step size performs best. At the end the group with the smallest step size has taken over. The total population size is distributed to the subpopulations according to their actual contribution to the search process. So the population sizes of the subpopulations vary over time whereas the total population size (denoted by SUM) remains constant.

Figure 3 shows an extended competition run with parameters similar to experiment 7. In order to get a better picture we used a consumption factor twice as high as in experiment 7. At the beginning the group performing fuzzy recombination performs best. At about generation 80 the group using mutation performs better and its population size increases. The total population size (SUM) decreases from 516 to 68.

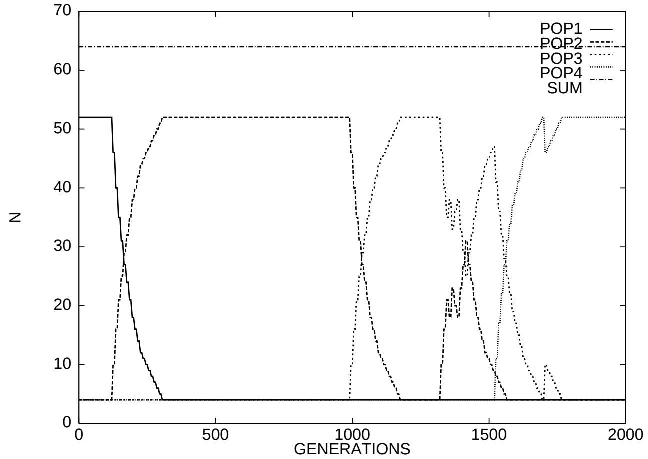

Fig. 2. Basic competition: development of population sizes of four competing subpopulations with different step sizes (see experiment 6).

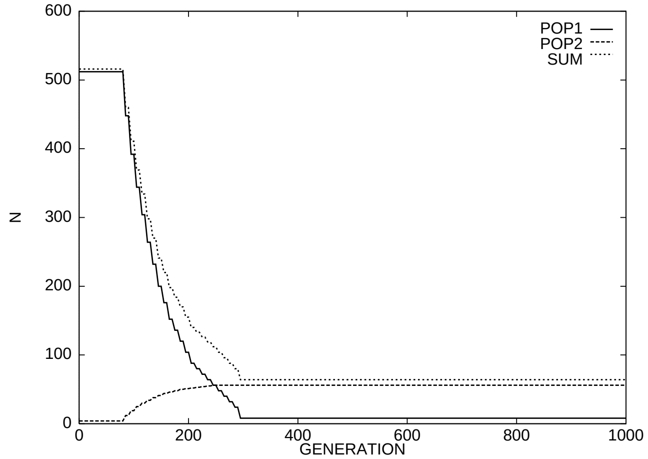

Fig. 3. Extended competition: development of population sizes of two competing subpopulations performing exploitating search (FR) and exploring search (BM^+) (similar to experiment 7 but with $\gamma_1 = 0.125$).

C. Results

The numerical results of all experiments are shown in Table 2. The difficulty of Rosenbrock's function for search is caused by the curved valley which has to be followed and a *second* region of attraction at $\vec{x} = (-1, 1, \ldots, 1)$. Many algorithms run into this point and are not able to follow the narrow valley out. The second difficulty can be avoided by starting at $\vec{x} = (0, 0, \ldots, 0)$. In this case a directed search will find the global optimum. Results for this experiment can be found in [6]. In contrast to this we initialize the population randomly in the search space. For this case the extended competition starting with a high population and fuzzy recombination (experiment 7) proved to be a very robust search principle. It is the only experiment which found the optimum in all of the 30 runs. The number of function evaluations needed is only about eight times of the best result of [6] which started at $\vec{x} = (0, 0, \ldots, 0)$. The experiments using the basic competition scheme (exp5 and exp6) show that it is more efficient than the standard BGA runs, but the robustness remains almost the same.

E	FE ($\times 10^3$) MEAN	SD	GEN	BF	F
1	–	–	–	15.659	30
2	118	41	14809	3.064	11
3	261	158	4072	1.196	9
4	–	–	–	34.746	30
5	71	44	4452	1.843	10
6	177	90	2758	2.086	8
7	196	24	2956	$1.0 \cdot 10^{-6}$	0

Table 2. Results for Rosenbrock's function ($n = 20$) with termination criterion $f_r(x) < 10^{-6}$ and initialisation: $\vec{x}_0 = \mathbf{U}([-5.12, 5.12])$ (30 runs). E : index of experiment, FE : function evaluation of successful runs, GEN: generations of successful runs, BF : best fitness, F : failed runs.

The real power of the extended competition becomes more impressive if a very difficult multimodal function like that of Fletcher and Powell is to be optimized. In Table 3 the setup of the three experiments with the function of Fletcher and Powell is shown. The first two experiments (8 and 9) are standard BGA runs without competition. Experiment 10 uses the extended competition model.

E	S	i	N_i^0	γ_i	OP FR	OP BM,BLR	T
8	1	1	512	1.0	FR	–	0.5
9	1	1	64	1.0	–	BM,BLR	0.25
10	2	1	512	0.0625	FR	–	0.5
		2	4	1.0	–	BM,BLR	0.25

Table 3. Experiments for Fletcher and Powell's function. The BGA-mutation $\mathsf{BM}(2\pi, 16, 0.0)$ is used.

Table 4 shows the results for dimension $n = 4$ and $n = 30$. In the table the average final best function value reached after 2000 generations is shown. The average of the euclidian distance to the optimum denoted by $\|\vec{x}\|$ shows that the breadth search (exp8) is able to locate a region which is closer to the optimum than the directed search (exp9) although the fitness value is worse. Note that the distance to the optimum is also larger for the extended competition (exp10) than for experiment 8. But the mean of the best fitness differs for $n = 4$ in 6 orders of magnitude! The extended competition (exp10) outperforms the standard BGAs using breadth search (exp8) or a directed search (exp9) in terms of best fitness achieved. For $n = 30$ the quality of the solution obtained by the extended competition is similar to the results in [1]. These results have been obtained without tuning by a standard setup. Further improvements for this difficult function are to be expected.

n	E	BF MEAN	SD	MIN	MAX	$\|x\|$
4	8	152.10	90.9000	7.82300	372.000	0.211
	9	34.14	187.0000	$3 \cdot 10^{-5}$	1024.000	4.177
	10	0.0008	0.0009	$2 \cdot 10^{-4}$	0.001	0.462
30	8	87390	26630	3656	137400	7.424
	9	14210	14060	935	56600	10.001
	10	2536	2763	28	11130	7.987

Table 4. Results for Fletcher and Powell after 2000 generations (30 runs). $\|x\|$ denotes the euclidian distance to the optimum.

VI. Conclusions

This paper focused on the adaptation of population sizes by competing subpopulations. The competition model dynamically allocated the number of trials to different search strategies by adapting the population sizes of subpopulations according to their actual contribution to the search. The basic competition model which keeps the total population size constant was extended to a competition model whose total population size is adapted during an optimization process. This extended competition scheme enables the use of search strategies needing different population sizes. Starting with a large subpopulation performing a breadth search and changing to a small subpopulation equipped with a directed search is extremely profitable for the optimization of multimodal problems.

The robustness of the extended competition model was demonstrated by applying BGA search strategies to two difficult functions. The real power of the extended competition scheme became obvious for multimodal functions.

References

[1] Thomas Bäck. *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, NY, January 1996.

[2] Thomas Bäck and Hans-Paul Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–24, 1993.

[3] J.M. Emlen. *Population Biology : The Coevolution of Population Dynamics and Behavior.* Macmillan Publishing Company, New York, 1984.

[4] R. Fletcher and M.J.D. Powell. A rapidly convergent descent method for minimization. *The Computer Journal*, 6:163–168, 1963.

[5] Terence C. Fogarty. Varying the probability of mutation in the genetic algorithm. In H. Schaffer, editor, *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 104–109, San Mateo, 1989.

[6] Nikolaus Hansen, Andreas Ostermeier, and Andreas Gawelczyk. On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In Larry J. Eshelman, editor, *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 57–63, San Francisco, CA, July 1995.

[7] Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer, Berlin, 1992.

[8] Heinz Mühlenbein and Dirk Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm: Continuous parameter optimization. *Evolutionary Computation*, 1(1):25–49, 1993.

[9] Heinz Mühlenbein and Dirk Schlierkamp-Voosen. The science of breeding and its application to the breeder genetic algorithm. *Evolutionary Computation*, 1(4):335–360, 1994.

[10] H. H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184, October 1960.

[11] Dirk Schlierkamp-Voosen and Heinz Mühlenbein. Strategy adaptation by competing subpopulations. In D. Davidor and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature 3*, pages 199–208, Jerusalem, October 1994. Springer.

[12] Hans-Michael Voigt and Heinz Mühlenbein. Gene pool recombination and the utilization of covariances for the breeder genetic algorithm. In *Proceedings of the 2nd IEEE International Conference on Evolutionary Computation*, pages 172–177, Perth, 1995. IEEE Press.

[13] Hans-Michael Voigt, Heinz Mühlenbein, and Dragan Cvetković. Fuzzy Recombination for the Breeder Genetic Algorithm. In Larry J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 104–111. Morgan Kaufmann, 1995.

appeared in: Proceedings of International Conference on
Evolutionary Computation (ICEC'96),
Nagoya, Japan, 1996, pp. 330-335.